



VAASAN AMMATTIKORKEAKOULU
UNIVERSITY OF APPLIED SCIENCES

Teemu Varpula

Bluetooth Low Energy IOT sovelluksessa

Kotlin sovellus Android Studiolla

Liiketalous
2020

TIIVISTELMÄ

Tekijä	Teemu Varpula
Opinnäytetyön nimi	Bluetooth Low Energy IOT sovelluksessa
Vuosi	2020
Kieli	suomi
Sivumäärä	34
Ohjaaja	Kenneth Norrgård

Tässä opinnäytetyössä aion tutkia miten Bluetooth Low Energy toimii ja miten hyödyllinen se on IOT ympäristössä. Ideana on tehdä Kotlin ohjelmointikielellä mobiilisovellus Android puhelimeen, jolla voidaan kontrolloida Bluetooth LE hehkulamppua ja lukea sen ominaisuuksia.

Käyn tässä työssä läpi Kotlin ohjelmointikieltä, Androidia, Android Studiota sekä Bluetooth Low Energyä. Otin selvää, kuinka käyttää Android Studion tarjoamia työkaluja ja rakentaa toimiva ja luotettava Bluetooth LE sovellus ja miten helppoa on käyttää Bluetooth LE:tä ja sen työkaluja.

Projektia tehdessä kävi ilmi, että Bluetooth LE on suhteellisen vähän käytetty ja osittain huonosti dokumentoitu teknologia, jonka käyttö tulee kasvamaan valtavasti tulevaisuudessa. Toisaalta Bluetooth LE:tä on helppo käyttää, kun sen idean oppii.

ABSTRACT

Author	Teemu Varpula
Title	Bluetooth Low Energy in IOT application
Year	2020
Language	suomi
Pages	34
Name of Supervisor	Kenneth Norrgård

This thesis investigated how Bluetooth Low Energy works and how useful it is in the IOT environment. The idea was to develop a mobile application using Kotlin programming language on an Android Phone that can control a Bluetooth LE light bulb and read its features.

In the thesis I examined Kotlin programming language, Android, Android Studio and Bluetooth Low Energy. I studied how to use the tools offered by Android Studio and how to build a working and reliable Bluetooth LE application. I also learned how easy it is to use Bluetooth LE profile and its tools.

During the project it became clear that Bluetooth LE is relatively little used and somewhat a poorly documented technology the use of which will increase tremendously in the future. On the other hand, Bluetooth LE is easy to use when its idea behind is internalized.

Keywords	programming tools, Bluetooth LE, IOT, mobile application, programming language
----------	--

SISÄLLYS

TIIVISTELMÄ

ABSTRACT

1	JOHDANTO	6
2	MIKÄ ON BLUETOOTH LOW ENERGY?.....	7
	2.1 Bluetooth LE:n erot Bluetoothiin	7
	2.2 Bluetooth LE:n tekniset ominaisuudet	8
	2.3 Bluetooth LE jokapäiväisessä elämässä ja IOT-laitteissa	11
3	KOTLIN	14
	3.1 Miksi valita Kotlin Javan sijaan	15
	3.2 Kotlinin tulevaisuus.....	16
4	ANDROID STUDIO	17
	4.1 Uusi projekti Android Studiolla	17
	4.2 Activityn valitseminen projektiin	18
	4.3.1 Käyttölupien määrittäminen Bluetooth LE sovellukselle.....	19
	4.3.2 Erikoiskäyttölupien määrittäminen on vaikeampaa	20
5	BLUETOOTH LE LAMPPUSOVELLUS.....	23
	5.1 Ohjelmointityökalujen haku Android Studiolla	24
	5.2 Bluetooth LE laitteiden haku	24
	5.3 Miten ylläpitää Bluetooth LE yhteyttä.....	25
	5.4 Bluetooth characteristic listaluettelo	26
	5.5.1 Datan lukeminen lampusta	27
	5.5.2 Datan lähettäminen lamppuun	28
	5.5.3 Notificationin vastaanottaminen ja lukeminen	28
	5.6 Lopullinen sovellus	30
6	LOPPUPÄÄTELMÄ PROJEKTISTA.....	31
	6.1 Android Studio kehitysympäristönä	31
	6.2 Sovelluksen toimivuus.....	31
	LÄHTEET.....	33

KUVIO- JA TAULUKKOLUETTELO

Kuvio 1. IndustryARC:in ennuste aktiivisten Bluetooth LE laitteiden määrästä maailmanlaajuisesti vuonna 2015. (Industry ARC 2015).	13
Kuvio 2. App Developer Magazinen teettämä kysely 2744 sovelluskehittäjälle siitä mihin he käyttävät Kotlinia. (App Developer Magazine 2018).	14
Kuvio 3. Esimerkki Javan ja Kotlinin eroista ohjelmointikielinä. (Android Authority 2019).	15
Kuvio 4. Kuvakaappaus Android Studion uuden projektin aloitussivulta.	17
Kuvio 5. Android Studion tarjoamat ohjelmointirajapintaversiot.	17
Kuvio 6. Kuvakaappaus activity valikosta Android Studiassa.	18
Kuvio 7. Manifest XML tiedosto.	19
Kuvio 8. Android Studion virheilmoitusloki.	20
Kuvio 9. Kuvakaappaus Android sovelluksen käyttöoikeudet-asetuksista.	21
Kuvio 10. Kuvakaappaus Android Studion koodista käyttöluvan tarkastukseen.	21
Kuvio 11. Kuvakaappaus Android puhelimen kysymysikkunasta käyttöluvan hyväksymiseen.	22
Kuvio 12. Työssä käytettävä Bluetooth LE hehkulamppu e-27 kannalla. (Pinterest PINkart.in Retails).	23
Kuvio 13. Kuvakaappaus Bluetooth LE:n kehitykseen vaadituista ominaisuuksista Android Studiassa.	24
Kuvio 14. Listaluetteloon löydetty Bluetooth LE hehkulamppu.	25
Kuvio 15. Kuvakaappaus sovelluksen löydetyistä characteristic arvoista.	26
Kuvio 16. Kuvakaappaus sovelluksen ikkunasta luettaessa characteristic arvo.	27
Kuvio 17. Kuvakaappaus sovelluksen lähdekoodista värin satunnaiseksi muuttamisesta.	28
Kuvio 18. Sovelluksen pääikkunat sekä hehkulamppu.	30

1 JOHDANTO

Mobiilisovelluksista ja IOT:stä kiinnostuneena halusin tehdä Bluetooth Low Energyä tutkivan opinnäytetyön ja kehittää sovelluksen, jolla voi hallita Bluetooth LE hehkulamppua. Päätin käyttää pääkielenä Kotlinia joka on paljon parempi ja modernimpi ohjelmointikieli kuin paljon moitittu Java. Työ on tehty Android Studiolla, jossa on hyvät työkalut Kotliniin ja Bluetooth LE:n kehittämiseen. Käytin sovelluksessa omaa Samsung Galaxy puhelinta, joka käyttää Android versiota 6.0.1 ja Bluetooth versiota 4.1.

Mitä hyötyä on Bluetooth Low Energystä ja millä lailla se on parempi kuin tavallinen Bluetooth? Miten helppoa on käyttää Android Studion työkaluja eri teknologioiden ja ohjelmointikielten kanssa? Vaikka Bluetooth LE ei ole mitenkään uusi teknologia markkinoilla, nimi on kuitenkin vähemmän tunnettu tavallisten käyttäjien keskuudessa ja sille kehitetään jatkuvasti uusia laitteita, joka tekee tästä ajankohtaisen aiheen opinnäytetööhön.

Bluetooth LE-lähetin onkin ollut kaikissa vuoden 2012 jälkeen myydyissä älypuhelimissa (Tendyck 2018).

Vaikka Bluetooth LE:llä on monta nimeä, aion käyttää pääsääntöisesti Bluetooth LE nimeä sen selkeyden vuoksi. En aio käydä läpi kaikkea mahdollista Bluetooth LE:n toiminnasta koska aihe on mittaamattoman laaja eikä se tieto ole oleellista koska ohjelmointirajapinnat tarjoavat valmiit työkalut Bluetooth LE:n kanssa toimimiseen.

Esittelen Bluetooth LE:tä ja sen teknisiä ominaisuuksia ja sen jälkeen käyn yksityiskohtaisesti läpi sovelluksen rakentamista alusta alkaen Android Studiolla.

2 MIKÄ ON BLUETOOTH LOW ENERGY?

Bluetooth Low Energy ilmestyi ensimmäisen kerran vuonna 2009 Bluetooth 4.0 version myötä. Alun perin sen kehitti Nokia vuonna 2006 nimellä Wibree ja nykyään sitä kutsutaan nimellä Bluetooth Smart, Bluetooth LE, BLE tai Bluetooth Low Energy.

Alun perin Bluetooth Low Energy kehitettiin halpavalmisteiseksi radiosignaalistandardiksi, joka on optimoitu kuluttamaan mahdollisimman vähän kaistatilaa ja energiaa sekä samalla tehdä sille helppo käyttöliittymä. (Townsend, Cufí, Davidson & Akiba 2014, 1)

Lyhyesti, Bluetooth LE on kuin normaali Bluetooth mutta optimoitu kuluttamaan mahdollisimman vähän energiaa.

2.1 Bluetooth LE:n erot Bluetoothiin

Bluetooth LE ei yritä korvata olemassa olevia teknologioita kuten WiFiä, Bluetoothia tai NFC:tä. Se yrittää olla paras omassa lajissaan ja isot teknologiayhtiöt kuten Apple ja Samsung ovat ottaneet sen laajasti käyttöön omissa laitteissaan, joten sillä on hyvin vahva jalansija tulevaisuudessa. Bluetooth LE:n asemaa vahvistaa tablettien, älypuhelimien, älykellojen ja tietokoneiden jatkuva käyttö ja kasvavat markkinat. Bluetooth LE on lisäksi halpa teknologia verrattuna muihin langattomiin tiedonsiirtoteknologioihin. Kuka tahansa voi tilata netistä sirujärjestelmän, jossa on Bluetooth LE lähetin ja hinta on huomattavasti halvempi verrattuna esim. WiFi:in tai GSM:ään. (Townsend ym. 2014, 1).

Perinteinen Bluetooth soveltuu paremmin tiettyihin tehtäviin kuten isojen datamäärien siirtoon ja Bluetooth LE puolestaan soveltuu lyhyisiin ja mahdollisesti useasti lähetettäviin pieniin datamääriin. Normaali Bluetooth on jatkuvasti yhteydessä laitteeseen, ellei yhteyttä katkaista, kun taas Bluetooth LE on yhteydessä pelkästään, kun se lähettää tai vastaanottaa dataa. Bluetooth LE on niin sanotussa nukkumistilassa, kun se ei ole yhteydessä laitteeseen. Jotkin paristoilla toimivat Bluetooth LE laitteet voivat kestää kuukausia tai jopa vuosia ennen kuin

virta loppuu ja akku pitää vaihtaa ja tämä ei ikinä olisi mahdollista normaalilla Bluetoothilla.

Yleensä Bluetooth LE kuluttaa puolet virtaa normaaliin Bluetoothiin verrattuna. (Mathieu 2019).

Aktiivisia Bluetooth LE laitteita voi teoriassa olla loputon määrä yhdistettynä käyttäjän päätelaitteeseen, joka on huima ero perinteisen Bluetoothin seitsemään laitteeseen. Oikea määrä vaihtelee Bluetooth LE version asennuksen mukaan, joten tarkkaa lukua ei voida antaa yleisesti Bluetooth LE:stä puhuttaessa.

Bluetoothin signaali yltää jopa 100 metrin päähän, kun taas Bluetooth LE riittää vain 50 metrin etäisyyteen, joka voikin selittää Bluetooth LE:n nopeampaa yhdistämistä ja tietojen siirtoa. Aikaviive laitteiden paritusten välillä on Bluetoothissa noin 100 ms ja Bluetooth LE:ssä 6 ms. (Summitdata, 2013)

2.2 Bluetooth LE:n tekniset ominaisuudet

Seuraavassa on luettelo tärkeimmistä Bluetooth LE:n toiminnan teknisistä ominaisuuksista ja nimityksistä. Aion käydä aiheet läpi lyhyesti ja selittää mitä ne tarkoittavat. Nämä ovat tärkeitä ominaisuuksia Bluetooth LE:n toiminnan kannalta ja tulen käymään näitä läpi lisää sovelluksen tekovaiheessa kappaleessa 5.

(1) GATT eli General Attribute Profile on yleinen määritelmä datan formatisointiin, pakkaamiseen, lähettämiseen ja vastaanottamiseen Bluetooth LE yhteyksien välillä. GATT:in tehtävä on hoitaa pelkästään sovelluksen ja käyttäjän datan siirto. Kaikkien Bluetooth LE yhteyksien on noudatettava tämän protokollan sääntöjä. (Townsend ym. 2014, 4).

(2) ATT eli Attribute Profile on myöskin Bluetooth LE tiedonsiirtoprotokolla. GATT käyttää ATT:ia kuljetusprotokollanaan datan lähettämiseen laitteiden välillä ja GATT onkin rakennettu ATT:in päälle. ATT perustuu dataan, joka lähetetään Bluetooth LE yhteyksien välillä, tätä datapakettia voidaan myös kutsua attribuutiksi. Attribuutteja voi kirjoittaa tai lukea. Attribuutit voi tulla käyttäjältä Bluetooth LE laitteeseen tai päinvastoin. (Townsend ym. 2014, 2).

(3) GATT-Client ja GATT-Server tarkoittaa eri rooleja, jota laitteilla on Bluetooth LE yhteyteen viitattaessa. Bluetooth LE:ssä jokainen laite on client tai server, tai laite voi olla molemmat samaan aikaan. Client pyytää dataa serveriltä ja server vastaa pyyntöön. (Townsend ym. 2014, 2).

Esimerkiksi jos otan yhteyttä Bluetooth LE lamppuun puhelimellani, silloin olen GATT client ja lamppu on GATT server. Toisaalta jos Bluetooth LE lamppu vaatisi omalta laitteeltani dataa käyttöönsä, silloin se on GATT client joka pyytää GATT serveriltä vastausta. Rooleilla ei ole niinkään väliä tavallisen käyttäjän kannalta, ne auttavat vain selkeyttämään ja ymmärtämään Bluetooth LE yhteyttä paremmin.

(4) GAP eli The Generic Access Profile on protokolla, jolla Bluetooth LE laitteet kommunikoivat protokollan alemmalla tasolla. GAP määrittää muutamia alemman tason yhteydensiirtoon liittyviä sääntöjä kuten laitteiden löytäminen, tietyn datan lähettäminen tai salatun yhteyden muodostaminen. Lyhyesti sanottuna GAP on vastuussa Bluetooth LE yhteyden toimivuudesta. (Townsend ym. 2014, 3).

(5) Characteristic on arvo, jota Bluetooth LE server laite tarjoaa. Characteristic voidaan lukea, siihen voi kirjoittaa tai sillä voi olla muita toiminnallisuuksia, joita kutsutaan propertyiksi. Eri Bluetooth LE laitteet tarjoaa eri määrän characteristic arvoja, joilla on eri propertyjä.

Esimerkkinä voi olla characteristic joka on laitteen nimi jolla voi olla pelkästään luku property joka tarkoittaa että käyttäjä voi lukea Bluetooth LE laitteen nimen characteristicistä mutta ei uudelleenkirjoittaa sitä. Jotkut characteristicit vaativat PIN koodin, että niitä voi käyttää. Alla käydään läpi Android Studion API 23 version tarjoamat 7 eri property arvoa:

- Read, characteristicin lukuominaisuus. Characteristicin arvon voi lukea.
- Write, characteristicin kirjoitusominaisuus. Characteristiciin voi kirjoittaa. GATT server lähettää vastauksen, jos kirjoitus characteristiciin onnistui. (Townsend ym. 2014, 4).

- Write without response, lähes sama kuin normaali kirjoitusominaisuus mutta vastausta ei tule serveriltä ollenkaan onko kirjoitus onnistunut vai ei. (Townsend ym. 2014, 4).
- Broadcast, erilaiset beacon laitteet voivat käyttää tätä esimerkiksi mainostuksessa. Characteristicin arvo voidaan lähettää mainostuspakettina. (Bhargava. 2017, 22)
- Notify, characteristicin ominaisuus lähettää automaattisesti päivittyviä notification viestejä. Tämän ominaisuuden voi laittaa päälle ja pois. Hyödyllinen useasti muuttuvissa arvoissa. (Bhargava. 2017, 72)
- Indicate, hyvin samantapainen kuin notify property mutta käyttäjän on hyväksyttävä viesti, kun se saapuu eikä GATT server lähetä uutta indicatea ennen kuin vanha on hyväksytty. (Bhargava. 2017, 72)
- Signed Write, hyvin samantapainen kuin normaali write mutta se tarjoaa mahdollisuuden käyttäjän todennukseen tai ns. allekirjoitukseen. (Bhargava. 2017, 60)

Valitettavasti dokumentaatio joistain propertyistä oli erittäin huonoa. Ajan myötä ne tulevat varmasti enemmän käytetyiksi mutta tärkeimmät characteristicin propertyt aloittelijoille ovat read, write, notify, indicate sekä broadcast. Tällä hetkellä nämä 5 propertyä on täysin riittäviä useimpien Bluetooth LE laitteiden kanssa työskentelyyn ja sovellusten rakentamiseen.

(6) UUID eli Universally unique identifier on 16 tavuinen luku- ja numerojono, joka on varmasti tai hyvin suurella varmuudella maailmanlaajuisesti ainutlaatuinen. UUID tunnustetta käytetään monissa sovelluksissa ja protokollissa mukaan lukien Bluetooth ja Bluetooth LE. (Townsend ym. 2014, 3).

Jokaisella characteristicillä on oma UUID tunniste. UUID on muotoa xxxxxxxx-0000-1000-8000-00805f9b34fb. Normaleissa Bluetooth LE laitteissa Characteristicin UUID on ennalta määrätty tehdasasetusten mukaan eikä sitä voi

muuttaa. Yhdistäminen ja kaikki viittaukset characteristiciin tapahtuu UUID:n kautta.

(7) Bluetooth Service on joukko characteristicjä. Esimerkkinä voi olla Bluetooth Service nimeltä Yleiset Tiedot ja sen sisällä voi olla eri characteristicjä kuten nimi, vuosiluku tai valmistajan tietoja. Jokainen Bluetooth Service sisältää myös oman UUID arvon.

Eri Bluetooth LE laitteet tarjoaa eri määrän Bluetooth Servicejä ja ne vaihtelevat eri laitevalmistajien toiveiden mukaan.

2.3 Bluetooth LE jokapäiväisessä elämässä ja IOT-laitteissa

Nykypäivänä Bluetooth LE laitteita on kaikkialla yritysten ja yksityishenkilöiden käytössä. Bluetooth LE on ollut merkittävässä asemassa IOT eli Internet Of Things vallankumouksen kanssa. (Nielsen 2017). Bluetooth LE soveltuu erittäin hyvin IOT laitteisiin alhaisen energiankulutuksen takia. Käyn läpi muutamia IOT ja Bluetooth LE yhdistettyjä laitteita seuraavassa luettelossa.

Bluetooth LE:n pääkäyttökohteita ovat:

1. Fitness laitteet, joissa on askelmittari, kalorilaskuri, kuntoiluohjelmia, sykemittari sekä muita yleishyödyllisiä toimintoja.
Hyvä esimerkki on Samsung Galaxy Fit ranneke, joka käyttää pelkästään Bluetooth LE versiota 5.0 lähettäessä dataa mobiilisovellukseen.
(Samsung 2016).
2. Terveystieteidenhuollossa käytettävät laitteet kuten sykemittarit, verenpainemittarit, lämpömittarit, unentestauslaitteet, kuulolaitteet sekä verensokerin mittauslaitteet.
MenaFN:n mukaan Bluetooth LE teknologia nostaa IOT terveystieteiden myyntiä vuoden aikana 29,9% vuoteen 2023 mennessä. (Guirdham, 2019).
3. Yritys- ja kotiautomaatiolaitteet: hehkulamput, lämpötilansäätimet, autotallin ovenavauslaitteet sekä erilaisten mittarien lukulaitteet.

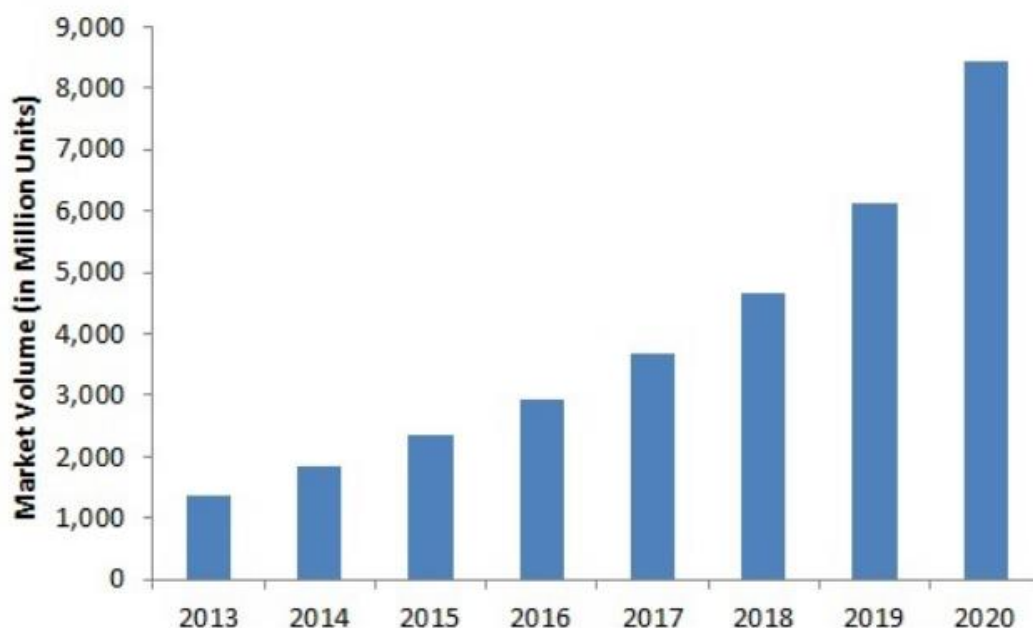
4. Turvallisuuslaitteet, jotka voivat jäljittää, paikantaa, tarkkailla tai kuunnella huonetta, ihmisiä tai eläimiä. Ne voivat olla kotona tai yrityksessä käytettäviä liiketunnistimia, kameroita tai hälyttimiä.
5. Smart Energy laitteet: Jääkaapit, pakastimet, pyykin- ja astianpesukoneet, kuivaajat, saunakiukaat tai kahvinkeitin. Nämä laitteet on optimoitu säästämään energiaa ja ne voivat säätää lämpötilaa automaattisesti eri sensorien avulla.

Vaikka nimi IOT viittaa internetiin, on Bluetooth LE silti merkittävässä roolissa IOT-laitteiden kanssa sen lähellä olevan tiedostonsiirtoteknologian takia. Bluetooth LE ei pysty lähettämään dataa eteenpäin palvelimille, joten se ei itse yksinään voi toimia IOT-laitteessa, joka tarvitsee internet yhteyden toimiakseen.

Toisaalta on olemassa IOT laitteita, jotka eivät tarvitse internetiä, jolloin Bluetooth LE on aivan riittävä vaihtoehto näiden laitteiden toimintaan.

2.4 Millaiselta näyttää Bluetooth LE:n tulevaisuus?

Vuodesta 2010 lähtien Bluetooth LE yhdistetyt laitteet on ollut tasaisessa kasvussa lisääntyneiden laitemäärien tullessa markkinoille ja kasvu ei näytä hidastuvan (Kuvio 1). Vuonna 2018 myytiin jopa 550 miljoonaa Bluetoothin sisältävää laitetta ja määrän odotetaan nousevan 850 miljoonaan vuoteen 2022 mennessä. Vuoden 2018 Bluetooth markkinaraportin mukaan 97% myydyistä Bluetooth siruista sisältää Bluetooth LE teknologian vuoteen 2022 mennessä. (Hollander 2018).



Kuvio 1. IndustryARC:in ennuste aktiivisten Bluetooth LE laitteiden määrästä maailmanlaajuisesti vuonna 2015. (Industry ARC 2015).

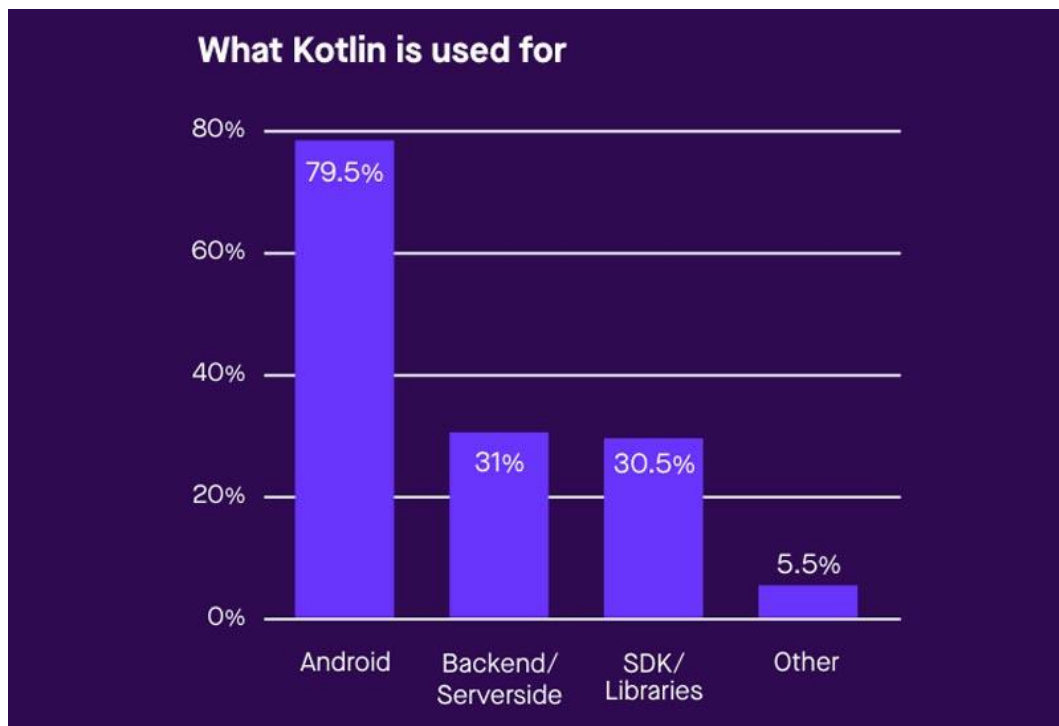
Yläpuolella oleva taulukko on hyvä esimerkki Bluetooth LE laitteiden määrän nopeasta kasvusta. Kun on toimiva tiedostonsiirtoprotokolla, joka on lähes jokaisessa puhelimessa vakiona, on ilmiselvää että vahva jalansija on taattu tulevaisuudessa. Lisäksi kun otetaan huomioon teknologian nopea kehitys sekä tarve uusille laitteille, joka on kasvanut räjähdysmäisesti niin Bluetooth LE laitteiden suosio on helppo selittää.

Tällä hetkellä Bluetooth äänentoistolaitteet käyttävät perinteistä Bluetoothia, joka johtuu enimmäkseen Bluetooth LE:n tavasta lähettää lyhyitä datamääriä. Tilanne tulee kuitenkin muuttumaan Bluetooth 5.0 version myötä, kaikki Bluetooth äänentoistolaitteet, jotka tukevat Bluetooth versiota 5.0 tulevat käyttämään Bluetooth LE:tä joka tulee vähentämään energian kulutusta ja säästämään akkua. Tässä tulee kestämään kuitenkin vielä hetki, esimerkiksi Applen AirPods kuulokkeet eivät käytä Bluetooth LE:tä vaan Bluetooth versiota 4.2. (Hoffman 2018).

3 KOTLIN

Kotlin on Tšekkiläisen JetBrains yhtiön vuonna 2011 perustama ohjelmointikieli. Se tukee Java Virtual Machine, Android, JavaScript sekä natiivia ohjelmistokehitystä. Muutamia isoja ohjelmistoja, jotka käyttävät Kotlinia on mm. Uber, Pinterest ja Evernote. (Kotlin)

Kotlin tituleeraa itseään staattisesti tyyplitetyksi ohjelmointikieleksi moderneille monialustaisille sovelluksille. (Harris 2018).



Kuvio 2. App Developer Magazinen teettämä kysely 2744 sovelluskehittäjälle siitä mihin he käyttävät Kotlinia. (App Developer Magazine 2018).

Kotlin ei yritä olla ainutlaatuinen omassa luokassaan vaan se perustettiin kymmenien vuosien ohjelmointikielten inspiroimana ja se lainaakin paljon ominaisuuksia muilta kieliltä kuten C#, Java sekä Scala. Kotlin on virallisesti Googlen tukema ja sitä ylläpitää Googlen lisäksi JetBrains yhtiön perustama Kotlin Foundation. (Android 2019).

3.1 Miksi valita Kotlin Javan sijaan

Java ohjelmointikieli perustettiin jo vuonna 1995 ja se on edelleen erittäin suosittu yrityksissä (Saurel 2019) eikä se todennäköisesti tule kuolemaan Kotlinin takia. Pääsyy Javan suosioon on miljoonat laitteet, jotka edelleen käyttävät ikivanhaa Javaa ja kielen vaihtaminen on hyvin kallista ja työlästä yrityksille.

KOTLIN

```
textView.setText("Hello World")
```

JAVA

```
TextView text = (TextView) findViewById(R.id.textView); text.setText("Hello World");
```

Kuvio 3. Esimerkki Javan ja Kotlinin eroista ohjelmointikielinä. (Android Authority 2019).

Kotlin lyhentää ohjelmointikoodia huomattavasti verrattuna Javaan (Kuvio 3) eikä sen kirjoittaminen ole niin tarkkaa kuin perinteisissä ohjelmointikielissä.

Esimerkiksi puolipilkkua ei tarvita koodirivin tekstin loppuun ollenkaan. Pääsyy käyttää Kotlinia Javan sijaan on Kotlinin yksinkertaisuus ja helppous sekä tietenkin se tosiasia että Kotlin kääntyy suoraan Java Virtual Machine laitteiden luettavaksi. Lisäksi Android Studio muuntaa Javan automaattisesti Kotliniksi ja päinvastoin jos käyttäjä haluaa, vaikkakin tämä ominaisuus välillä takkuili omassa käytössä.

Javassa muuttujan arvon voi asettaa null arvoon, joka myöhemmin voi aiheuttaa päänsäivää koska ohjelma yrittää käyttää null arvoa, joka on mahdotonta ja ohjelma kaatuu. Tämä ei ole Kotlinissa mahdollista koska muuttujat ovat vakiona ei-nullattavia. Toisin kuin Java, Kotlin ei edes yritä koota ohjelmaa, jos se huomaa mahdollisen null arvon, joka kaataisi ohjelman. (Educba 2018).

Toisaalta jos käyttäjä välttämättä haluaa asettaa muuttujan arvon nulliksi Kotlinissa, se onnistuu käyttämällä kahta huutomerkkiä muuttujan tyyppin perässä, esimerkiksi `Int` muuttuu `Int!!`. Tämä vakuuttaa, että muuttuja ei ole null ja ohjelman voi ajaa mutta ohjelma voi silti kaatua, jos muuttujaa koettaa käyttää kun se on

null. Toinen tapa on käyttää kysymysmerkkiä, jolloin se muuttuu Int? arvoksi. Tämä kertoo ohjelmalle, että muuttujalla on mahdollisuus olla null arvo.

3.2 Kotlinin tulevaisuus

Itse uskon että Kotlin tulee kasvamaan suosituksi tulevaisuudessa (Kuvio 2) ja on ehdottomasti oppimisen arvoinen kieli varsinkin mobiilisovelluksien kehityksessä. Java on yksinkertaisesti työläs kieli käyttää, vaikka se on silti erittäin suosittu.

Tällä hetkellä Kotlin käyttäjiä on huomattavasti vähemmän kuin Javalla, esimerkiksi Stack Overflow sivustolla oli 26 000 kysymystä Kotlin aiheesta vuonna 2019. Samana vuonna Java aiheisia kysymyksiä oli huimat 1,5 miljoonaa. (Jackowski 2019).

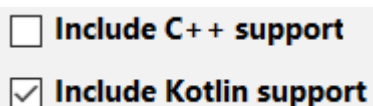
Kaikkien suosituimmista ohjelmointikielistä Kotlin nousi sijalle 27 vuonna 2018. Edellisenä vuonna se oli vielä sijalla 65 joka tekee siitä toiseksi nopeiten kasvavan ohjelmointikielen Swift'in jälkeen. (Rayome 2018).

4 ANDROID STUDIO

Android Studio on Googlen vuonna 2013 kehittämä ohjelmistokehitysalusta. Pääohjelmointikielet Android Studiolla ovat Java, Kotlin sekä C++ mutta siihen voi ladata laajennuksia ja täten hankkia paljon lisää ohjelmointikieliä. Sen voi ladata ilmaiseksi osoitteesta <https://developer.android.com/studio/>. Aion käydä läpi projektin tekemistä Android Studion normaaliversiossa ilman laajennuksia.

4.1 Uusi projekti Android Studiolla

Uuden projektin voi aloittaa New Project valikosta. Ensimmäisellä sivulla on mahdollisuus valita C++ tai Kotlin kielituki. Molemmat voi valita tai jättää valitsematta. Jos ei valitse kumpaakaan, on ohjelmointikieli Java. Valitsen Kotlin tuen koska aion käyttää sitä sovelluksessa (Kuvio 4).



Kuvio 4. Kuvakaappaus Android Studion uuden projektin aloitussivulta.

Projektille on annettava nimi ja tallennuskansioipaikka. Sen jälkeen on valittava minimi SDK eli Software Development Kit versio. Tämä vaihtelee sen mukaan mille laitteille sovellusta halutaan tehdä. Käytän tässä työssä Android 6.0.1 versiota omassa puhelimessa, joten valitsen SDK version API 23: Android 6.0 (Marshmallow).



Kuvio 5. Android Studion tarjoamat ohjelmointirajapintaversiot.

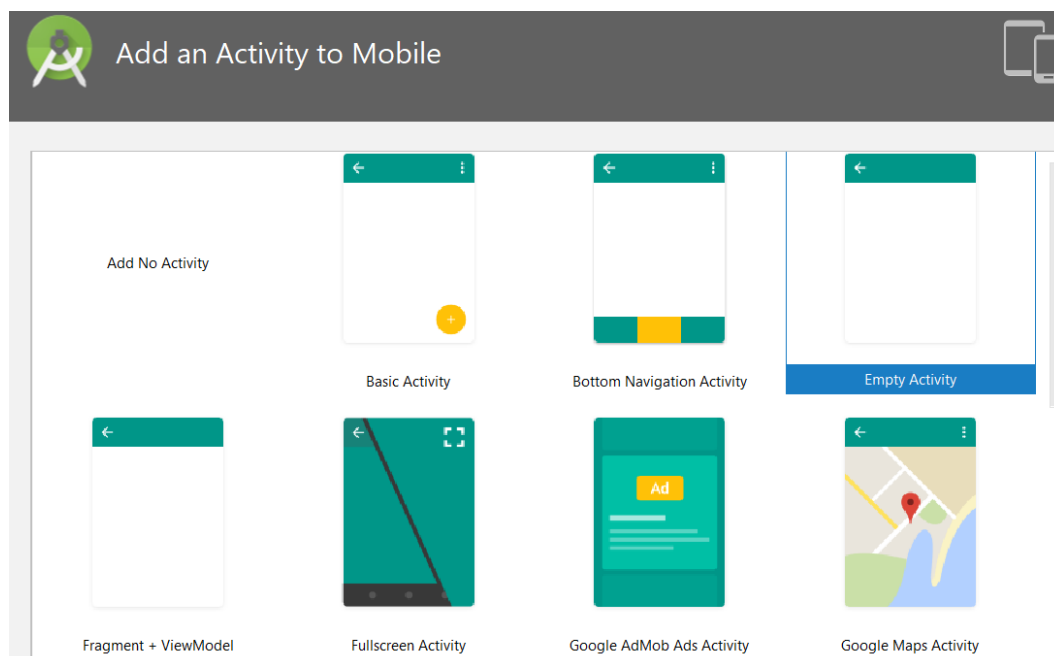
API eli Application Programming Interface on ohjelmointirajapinta Android Studiossa, joka tarjoaa työkalut ohjelmistokehitykseen. API 28 olisi tämän hetken uusin täysi versio (Kuvio 5) ja se tarjoaa paljon enemmän ominaisuuksia kuin API

23 mutta koska uudet ominaisuudet ei toimisi omalla puhelimellani, en aio käyttää sitä. Lisäksi uusimman API:n käyttäminen rajoittaa valtavasti kohdeyleisöä, jos sen haluaa ladata Google Play sovelluskauppaan muiden ladattavaksi.

API 28, Android 9.0 Pie tukee alle 1% saatavina olevista Android laitteista, kun taas API 23 tukee jopa 62,6%. Nämä luvut tietenkin muuttuvat jatkuvasti, kun Android versioita päivitetään ja uusia laitteita tulee markkinoille. Tällä hetkellä API 15, Android versio 4.0.3 tukee 100% kaikista laitteista ja teoriassa sillä voisi tehdä sovelluksia kaikille Android laitteille. On kuitenkin pidettävä mielessä, että työkalut ovat hyvin rajoittuneita vanhemmilla versioilla.

4.2 Activityn valitseminen projektiin

Seuraavaksi valitaan mahdollinen activity sovellukselle. Empty Activity on nopea ja helppo valinta koska ohjelma luo automaattisesti MainActivity päätiedoston sekä Layout tiedoston, joka on käyttäjän näkemä sovellusikkuna. MainActivity tiedosto on päätiedosto missä koodi aloitetaan ja mitä käyttäjä ei näe. Layout tiedosto on kaikki mitä käyttäjä näkee ja se on XML muotoa.



Kuvio 6. Kuvakaappaus activity valikosta Android Studiossa

Ohjelma tarjoaa lukuisia hyödyllisiä activity vaihtoehtoja (Kuvio 6). Basic Activity lisää automaattisesti Floating Action Buttonin ja Coordinator Layoutin, joka

käytännössä tarkoittaa valmista pohjaa tietyn tyyppiselle sovellukselle. Ohjelma määrittää pienen napin ikkunan oikeaan alalaitaan sekä se tekee automaattisesti säännöt, kuinka elementit järjestyvät käyttäjän näkemässä Layout tiedostossa.

Tämän kaiken voisi tietenkin tehdä tyhjässä activityssä itse lisäämällä koodin Layout tiedostoon, se säästää vain aikaa. Valmiit activityt muokkaavat koodia valmiiksi ja niistä voi saada hyvän pohjan eri sovelluksille.

4.3.1 Käyttölupien määrittäminen Bluetooth LE sovellukselle

Lukuisat toiminnot kuten esimerkiksi WiFi, Bluetooth LE tai NFC tarvitsevat käyttöluvan Android Studioissa toimiakseen sovelluksessa. Tiedostovalikossa on Manifest XML tiedosto, jonne on määriteltävä luvat mitä haluaa käyttää, tai muuten ne eivät toimi sovelluksessa.

Tässä sovelluksessa aion lisätä BLUETOOTH, BLUETOOTH_ADMIN, ACCESS_COARSE_LOCATION sekä ACCESS_FINE_LOCATION luvat koska Bluetooth ja Bluetooth LE tarvitsee ne toimiakseen (Kuvio 7).

Android versio yhdeksälle ja sitä aiemmille versioille riittää pelkästään ACCESS_COARSE_LOCATION, joka on riittävä sovelluksen toimintaan, jolloin ei tarvitse valita ACCESS_FINE_LOCATION lupaa. Toisaalta sovelluksen toiminnan kannalta ei ole mitään haittaa, jos valitsee molemmat.

```
<uses-permission android:name="android.permission.BLUETOOTH"/>
<uses-permission android:name="android.permission.BLUETOOTH_ADMIN"/>
<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION"/>
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION"/>

<uses-feature android:name="android.hardware.bluetooth_le" android:required="false"/>
```

Kuvio 7. Kuvakaappaus Manifest XML tiedosto

Viimeisellä rivillä kuvassa 6 on uses-feature android.hardware.bluetooth_le ominaisuus joilla voidaan määrittää toimiiko sovellus vain Bluetooth LE:tä tukevilla laitteilla. Jos ominaisuutta ei lisätä Manifest tiedostoon, ei koodilla voi tarkistaa tukeeko puhelin Bluetooth LE ominaisuutta ja täten vaatia sitä. Jos android:required= arvon asettaa todeksi, silloin sovellus ei toimi laitteilla joissa ei ole Bluetooth LE ominaisuutta. Tämän arvon muuttaminen on aika yhdentekevää

sovelluksen kannalta koska sen pystyy kuitenkin tarkistamaan myöhemmin koodissa, jos haluaa.

4.3.2 Erikoiskäyttölupien määrittäminen on vaikeampaa

Alkaen Android 6.0:sta, API versio 23:sta, erikoiskäyttölupien määrittäminen Manifest tiedostoon ei enää yksistään riitä, vaan sovellukselle on myös annettava lupa käyttää ominaisuuksia. Tämä uudistus pätee myös vanhoihin API versioihin. Jos ominaisuudella oli lupa toimia sovelluksessa aiemmin, ei ole mitään takeita että se toimii jatkossakin. (Android Developers).

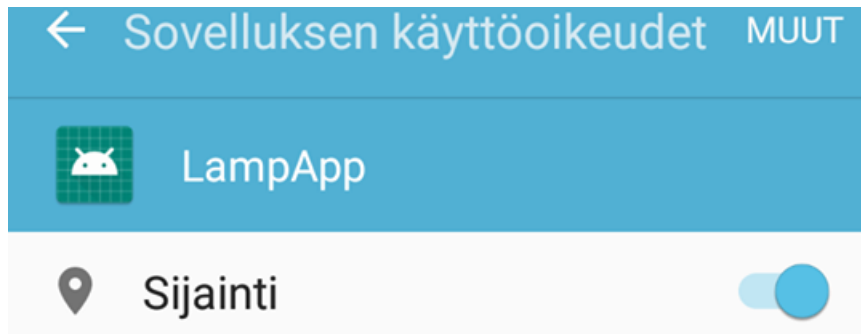
Tämän ongelman voi korjata kahdella tapaa, vaatia käyttäjää sallimaan sovelluksessa käytettävä ominaisuus kuten sijainti tai ehdottaa että käyttäjä itse laittaa luvat päälle Android asetusten sovellusluettelosta. Vaikka käyttäjä asettaa sijainnin päälle ennen sovelluksen käynnistämistä, ei se yksinään anna sovellukselle lupaa käyttää sitä.

```
java.lang.SecurityException: Need ACCESS_COARSE_LOCATION or ACCESS_FINE_LOCATION permission to get scan results
    at android.os.Parcel.readException(Parcel.java:1620)
    at android.os.Parcel.readException(Parcel.java:1573)
    at android.bluetooth.IBluetoothGatt$Stub$Proxy.startScan(IBluetoothGatt.java:890)
    at android.bluetooth.le.BluetoothLeScanner$BleScanCallbackWrapper.onClientRegistered(BluetoothLeScanner.java:375)
    at android.bluetooth.IBluetoothGattCallback$Stub.onTransact(IBluetoothGattCallback.java:56)
    at android.os.Binder.execTransact(Binder.java:453)
```

Kuvio 8. Kuvakaappaus Android Studio virheilmoitus logi.

Android Studio antaa selvän virheilmoituksen logiin jos käyttölupaa ei ole annettu vaikka se olisikin määritelty Manifest tiedostossa (Kuvio 8). Ohjelma ei anna sen parempaa neuvoa vaan internetistä löytää ratkaisun tähänkin ongelmaan.

Helpointa sovelluskehittäjän kannalta olisi vaatia käyttäjää itse laittamaan käyttöluvat päälle puhelimessaan, oli se kuinka vaikeaa tahansa. Omassa puhelimessani löysin sen pitkän etsinnän jälkeen asetuksista sovelluksen tiedoista.



Kuvio 9. Kuvakaappaus Android sovelluksen käyttöoikeudet-asetuksista.

Sovellukseni näyttää yhden ainoan vaaditun luvan eli sijainnin (Kuvio 9). Tämä johtuu siitä, että lisäsin Manifestiin `COARSE_LOCATION` sekä `FINE_LOCATION` luvat, jolloin sovellus osaa näyttää nämä erikoisluvut luettelossa. Bluetooth LE tarvitsee nämä luvat toimiakseen mutta normaali Bluetooth ei. Luettelossa ei myöskään näy Bluetoothia koska se ei ole erikoislupa vaan siihen riittää Manifest tiedoston muokkaaminen. Jos laitan tämän luvan päälle, sovelluksen Bluetooth LE ominaisuus toimii.

Yksikään toimiva sovellus ei tyytyisi tähän ratkaisuun koska käyttäjän ei voi olettaa tietävän jokaista pientä niksia puhelimessa eikä käyttäjän kuulu nähdä näin paljon vaivaa luvaa päälle laittoon. Onneksi Android Studio tarjoaa työkalut tämän asetuksen helpottamiseen.

4.3.3 Käyttölupaongelman ratkaisu

Yksinkertainen lisäys koodiin tarkistaa onko lupa `ACCESS_COARSE_LOCATION`:iin annettu (Kuvio 10). Jos lupaa ei ennestään ole niin sovellus vaatii sitä ja avaa yksinkertaisen kysymysikkunan (Kuvio 11). Jos käyttäjä sallii sovelluksen käyttää ominaisuutta, lupa menee päälle Androidin asetuksissa eikä sitä ei tarvitse enää ikinä kysyä uudelleen, ellei käyttäjä muuta asetusta itse.

```
if (ContextCompat.checkSelfPermission(this,
    Manifest.permission.ACCESS_COARSE_LOCATION)
    != PackageManager.PERMISSION_GRANTED) {
    ActivityCompat.requestPermissions(this,
        arrayOf(Manifest.permission.ACCESS_COARSE_LOCATION), 0)
```

Kuvio 10. Kuvakaappaus Android Studion koodista käyttöluvan tarkastukseen.

Sana this viittaa koodissa luokkaan, jonka sisällä koodi on, tässä tapauksessa MainActivity. CheckSelfPermission sekä requestPermissions ovat Googlen itse kehittämiä toimintoja, joita voi käyttää vain Androidia kehittäessä ja niiden käyttö vaihtelee eri ohjelmointikielten mukaan.



Kuvio 11. Kuvakaappaus Android puhelimen kysymysikkunasta käyttöluvan hyväksymiseen.

Kun tämä kaikki on tehty, voidaan vihdoinkin aloittaa sovelluksen teko MainActivity ja layout tiedoston muokkaamisella. Alussa voi olla järkevää ajaa tyhjä sovellus ja tarkistaa että kaikki toimii oikein. Kun sovellus ajetaan ensimmäisen kerran, se asentuu puhelimeen automaattisesti.

Android Studiolla voi tehdä sovelluksia oikealle puhelimelle tai virtuaaliympäristöön simuloidulle alustalle kuten tabletille tai puhelimelle. Kun puhelin on yhdistetty tietokoneeseen niin Android Studio ilmoittaa mallin nimen ja laitteen käyttämän API version.

5 BLUETOOTH LE LAMPPIUSOVELLUS

Tässä sovelluksessa yhdistän Magic Blue Bluetooth LE lamppuun (Kuvio 12) joka käyttää Bluetooth versiota 4.0. Paketissa ei ollut dokumentaatiota lampun toiminnasta vaan valmistaja tarjoaa omaa sovellusta ladattavaksi lampun käyttämiseen. Lamppu on valmis käyttöön, kun sen asettaa paikoilleen lampun pidikkeeseen. Kun valon laittaa päälle, lamppu muistaa viimeksi asetetun väriarvon ja käyttää sitä. Bluetooth LE toimii lampussa automaattisesti aina kun se on päällä.



Kuvio 12. Työssä käytettävä väriä vaihtava Bluetooth LE hehkulamppu e-27 kannalla. (Pinterest PINKart.in Retails).

Huomioitavaa Bluetooth LE lampussa on se, että sitä ei tarvitse parittaa Bluetooth pariaksi käyttäkseen sitä, se on täysin vapaaehtoista ja se toimii paritettuna tai ilman paritusta. Lisäksi siinä ei ole minkäänlaista suojausta kuten PIN-koodia, joka tarkoittaa että kuka tahansa tarpeeksi lähellä oleva pystyy kontrolloimaan sitä. Tämä ei kuitenkaan ole yleistä Bluetooth LE laitteille, Bluetooth LE sisältää mahdollisuuden asettaa PIN-koodin laitteiden paritukseen tai yhdistettäessä characteristic arvoon. Niiden käyttö vaihtelee laitevalmistajien mukaan.

Bluetooth ja Bluetooth LE onkin saanut huomiota huonosta tietoturvasta. Molemmat teknologiat ovat haavoittuvaisia salakuuntelulle ja MIM eli man in the middle hyökkäyksille. Turvallinen Bluetooth LE yhteys on suojattu salakuuntelulta mutta MIM hyökkäyksien estoon tarvitaan vahva paritus. (Kacherovska 2019). Jos joku ulkopuolinen yhdistäisi Bluetooth LE lamppuun sen ollessa päällä, ei kyseessä

kuitenkaan olisi salakuuntelu tai MIM hyökkäys koska laitetta ei ole edes tarkoitettu suojattavaksi eikä data sisällä mitään arkaluontoista tietoa.

5.1 Ohjelmointityökalujen haku Android Studiolla

Tyypillisen ohjelmistokehitysalustan mukaan on ensin tuotava ominaisuudet ohjelmiston käyttöön ja Android Studiossa se tapahtuu import sanalla. Android Studiossa on valtava määrä ominaisuuksia, joita mobiilisovellus tarvitsee toimiakseen ja Bluetooth LE:llä on myös omat ominaisuudet, joita se käyttää.

```
import android.bluetooth.BluetoothAdapter
import android.bluetooth.BluetoothGatt
import android.bluetooth.BluetoothGattCallback
import android.bluetooth.BluetoothGattDescriptor
import android.bluetooth.BluetoothGattService
import android.bluetooth.BluetoothManager
import android.bluetooth.BluetoothProfile
import android.bluetooth.BluetoothGattCharacteristic
import android.content.BroadcastReceiver
import android.bluetooth.le.ScanCallback
import android.bluetooth.le.ScanResult
```

Kuvio 13. Kuvakaappaus Bluetooth LE:n kehitykseen vaadituista ominaisuuksista Android Studiossa.

Jokainen tuotu työkalupaketti lisää lukemattomia ominaisuuksia ja ne onkin jaettu hyvin tarkasti eri paketteihin (Kuvio 13). Tämä mahdollistaa työskentelyn hyvin yksityiskohtaisten apuvälineiden kanssa ja suuri määrä työkaluja hidastaa sovellusta turhaan, jos niitä kaikkia ei käytä.

5.2 Bluetooth LE laitteiden haku

Alussa teen listaluettelon, joka hakee ja näyttää lähellä olevat Bluetooth LE laitteiden nimet ja osoitteet (Kuvio 14). Normaaleja Bluetooth laitteita listaan ei tule ollenkaan. Jos laitteella ei ole nimeä se on automaattisesti null. Käytän Android Studion yksinkertaista listaluetteloa eli ListView elementtiä. Listaluettelo on helppoa lisätä helpposti objekteja ja niihin voi asettaa kuuntelijan, joka kuuntelee koska objekteista klikataan.

Nimi: LEDBLE-7863109D
Osoite: F8:1D:78:63:10:9D

Kuvio 14. Listaluetteloon löydetty Bluetooth LE hehkulamppu.

Jos listaluettelossa olevasta objektista eli Bluetooth LE laitteesta painaa, avautuu uusi Activity ja puhelin yrittää yhdistää laitteeseen. Tämä yhdistäminen ei käytännössä vielä tee mitään ja se voi epäonnistua, jos laitteeseen ei saa yhteyttä jostain syystä. Se ei myöskään lähetä dataa yhdistämisvaiheessa.

On kuitenkin tärkeää muistaa työvaiheet, kun Bluetooth LE laitteesta klikataan. Ensin pitää lopettaa laitteiden skannaus, jos se on käynnissä tai muuten sovellus voi hidastua toimintakelvottomaksi tai kaatua. Tämän jälkeen otetaan valitun laitteen nimi sekä Bluetooth osoite ja lähetetään ne eteenpäin uudelle activitylle.

5.3 Miten ylläpitää Bluetooth LE yhteyttä

Erittäin tärkeä ominaisuus Bluetooth LE laitteeseen yhdistettäessä on nähdä yhteyden tila ja kuunnella päivittyviä tapahtumia. Jos yhteys katkeaa, on estettävä tai poistettava joitain ominaisuuksia käytöstä tai muuten sovellus kaatuu, jos käyttäjä yrittää lähettää dataa olemattomaan laitteeseen.

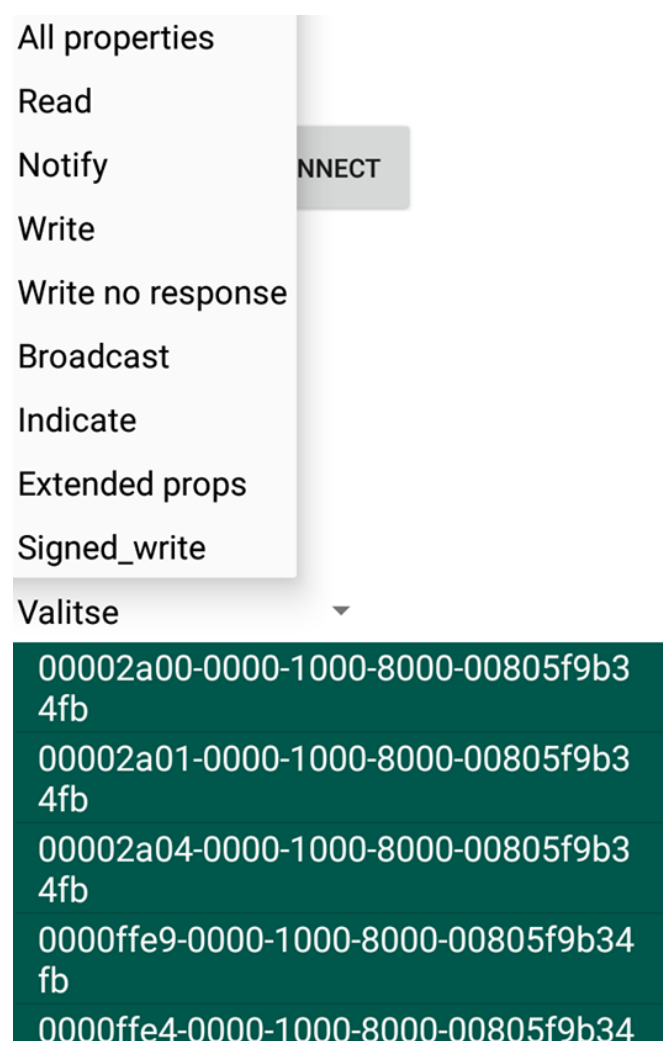
Android Studiossa on työkalu nimeltä BroadcastReceiver joka kuuntelee tapahtumia Bluetooth LE laitteelta. Nämä tapahtumat ovat:

- ACTION_GATT_CONNECTED = Yhteys onnistui.
- ACTION_GATT_DISCONNECTED = Yhteys katkesi.
- ACTION_GATT_SERVICES_DISCOVERED = Bluetooth LE laitteesta löydettiin Servicejä ja characteristicjä. Nämä voidaan nyt tuoda näkyviin. Ne voidaan erotella Service tai characteristic listana tai näyttää molemmat samaan aikaan.
- ACTION_DATA_AVAILABLE = Dataa on luettu ja se voidaan näyttää. Tämä tapahtuma on silloin kun characteristiciä luetaan onnistuneesti tai notification on lähettänyt dataa.

Tarkemmin sanottuna nämä tapahtumat tulevat BluetoothGattCallback työkalulta, jolla on mahdollisuus kuunnella jos yhteyden tila muuttuu. Yksinkertaisesti sanottuna nämä kaikki tapahtumat tulevat Bluetooth Serveriltä ja kutsuja on Bluetooth Client eli tässä tapauksessa sovellukseni.

5.4 Bluetooth characteristic listauettelo

Bluetooth LE laitteen tarjoamat characteristicit saa haettua onnistuneen yhteydenoton jälkeen. Niitä voi hakea myös erikseen niiden ominaisuuksien mukaan, joka voi helpottaa niiden kanssa työskentelyä.



Kuvio 15. Kuvakaappaus sovelluksen löydettyistä characteristic arvoista.

Erottelin characteristicit ominaisuuksien mukaan (Kuvio 15) ja hain niiden UUID arvon listauetteloon. Kaikilla ominaisuuksilla on oma Int lukuarvo tai ne voi

erotella tekstimuodon perusteella, esimerkiksi read lukuarvo on 2 ja write on 8. Nämä lukuarvot helpottavat niiden erottelua ja niiden kanssa työskentelyä.

Näiden characteristic arvojen erottelu on kätevää koska niillä on eri ominaisuuksia, esimerkiksi read characteristicin arvon voi yrittää lukea, kun taas write arvoa ei voi.

Tässäkin on hyvä muistettava, että jos yhteys katkeaa, ei characteristicien kanssa voi työskennellä ja on tehtävä tarvittavat toimenpiteet sovelluksen kaatumisen ehkäisemiseksi.

5.5.1 Datan lukeminen lampusta

Characteristic arvon luku on hyvin helppoa. Se tehdään komennolla `readCharacteristic(Characteristic)` ja data saapuu aina bytearray muodossa. Se on helppo formatisoida hexadesimaalimuotoon ja sen jälkeen luettavaan muotoon.

State: Connected

Data: LEDBLE-7863109D

4C 45 44 42 4C 45 2D 37 38 36 33 31
30 39 44 00

Kuvio 16. Kuvakaappaus sovelluksen ikkunasta luettaessa characteristic arvo.

Tässäkin tapauksessa `BluetoothGattCallback` kuuntelija odottaa koska dataa luetaan ja ilmoittaa siitä eteenpäin, kun data on vastaanotettu. Metodi, joka kuuntelee saapuvaa dataa, on nimeltään `onCharacteristicRead`.

Huomasin että data, joka voidaan lukea eri characteristic arvoista on usein virheellistä tai sitä ei voida tulkata luettavaan muotoon. Tämän takia on tärkeää sovelluksen kehitysvaiheessa lukea se myös heksadesimaalimuodossa koska silloin kaikki merkit ovat tarkasti nähtävillä eikä tulkitsemattomia merkkejä tai tyhjiä välejä ilmesty. (Kuvio 16).

5.5.2 Datan lähettäminen lamppuun

Jos dataa haluaa lähettää onnistuneesti Bluetooth LE laitteeseen, on se tehtävä oikeaan characteristic arvoon oikeassa muodossa. Datan lähettäminen Bluetooth ja Bluetooth LE yhteyden kautta on tehtävä aina ByteArray muodossa.

Datan lähettäminen on suhteellisen yksinkertaista komennolla `writeCharacteristic(characteristic)` jolloin data lähetetään kirjoittamalla se characteristicin arvoon.

```
val red = String.format("%02X", 0xFFFFFF and Random().nextInt(256))
val green = String.format("%02X", 0xFFFFFF and Random().nextInt(256))
val blue = String.format("%02X", 0xFFFFFF and Random().nextInt(256))
val originalString = "56" + red + green + blue + "00F0AA"
mBluetoothLeService!!.sendDataToBleDevice(mySpecialCharacteristic, originalString)
```

Kuvio 17. Kuvakaappaus sovelluksen lähdekoodista värin satunnaiseksi muuttamisesta.

Tässä tapauksessa lähetän dataa Bluetooth LE lamppuun, joka hyväksyy dataa muodossa `56f12345f12345` eli `56` + heksadesimaalivärikoodi + heksadesimaalivärikoodi (Kuvio 17). Se on täysin lampun valmistajan päätös lähettää dataa tässä muodossa tiettyyn characteristic arvoon. `56` on heksadesimaaliarvona kirjain `V`, joka voi viitata sanaan `value`. Sen jälkeen kirjoitetaan RGB värikoodi, joka määrää lampun värin. Toinen RGB värikoodi tämän jälkeen on muuhun tarkoitukseen kuten lampun värilämmön säätelyyn. (Shaked 2016).

Teoriassa lamppu tarjoaa $16\,777\,216$ eri väriarvoa eli 256 potenssiin 3 , jokaisella päävärillä voi olla 256 eri arvoa. Käytännössä niin monen värin erottaminen on tietenkin mahdotonta mutta teoriassa se toimii koska kaikki värikoodit ovat erilaisia.

5.5.3 Notificationin vastaanottaminen ja lukeminen

Aiemmin kappaleessa 2.2 kävin läpi Bluetooth LE Characteristic ominaisuuksia. Yksi tärkeimmistä ominaisuuksista on notification joka on hyvin tehokas tapa kuunnella saapuvaa dataa. Esimerkiksi sykemittari voi lähettää mitatun sykkeen

arvon aina kun se muuttuu tai tietyn ajanjakson välein. Notification nopeuttaa datan automaattista vastaanottamista koska yhteyttä ei tarvitse avata ja sulkea joka kerta eikä samaa characteristic arvoa tarvitse lukea uudelleen kerta toisensa jälkeen.

Notificationin idea on hyvin yksinkertainen mutta sen toteuttaminen oli yllättävän vaikeaa. Ensiksi annetaan yksinkertainen komento `setCharacteristicNotification(characteristic, true)`. Tämä ei riitä vielä vaan se yksinkertaisesti asettaa notificationin päälle client puolella käyttäjän laitteella, mutta ei server laitteella.

On otettava apuun CCCD eli Client Characteristic Configuration Descriptor. Tämä on ominaisuus, joka on erittäin hyödyllinen useimmissa Bluetooth LE käyttötapauksissa. Se toimii kuin kytkin, sillä voi laittaa päälle server laitteen lähettämät päivitykset client laitteelle. Sitä käytetään notification ja indication arvojen lähettämisessä ja vastaanottamisessa. (Townsend ym. 2014, 4).

CCCD:llä on myös oma UUID, joka on 00002902-0000-1000-8000-00805f9b34fb. Tämän UUID:n descriptor kuvaukseen kirjoitetaan `ENABLE_NOTIFICATION_VALUE`, joka asettaa notification ilmoitukset päälle server laitteessa. Bluetooth server laite lukee tämän descriptor arvon ja asettaa ilmoitukset päälle.

Kaiken tämän jälkeen laite on valmis vastaanottamaan notification ilmoituksia. Jos uusi ilmoitus saapuu, `BluetoothGattCallback` työkalu ilmoittaa siitä ja kertoo notificationin viestin hexadesimaalimuodossa. Metodi, joka kuuntelee saapuvaa notification dataa on nimeltään `onCharacteristicChanged`.

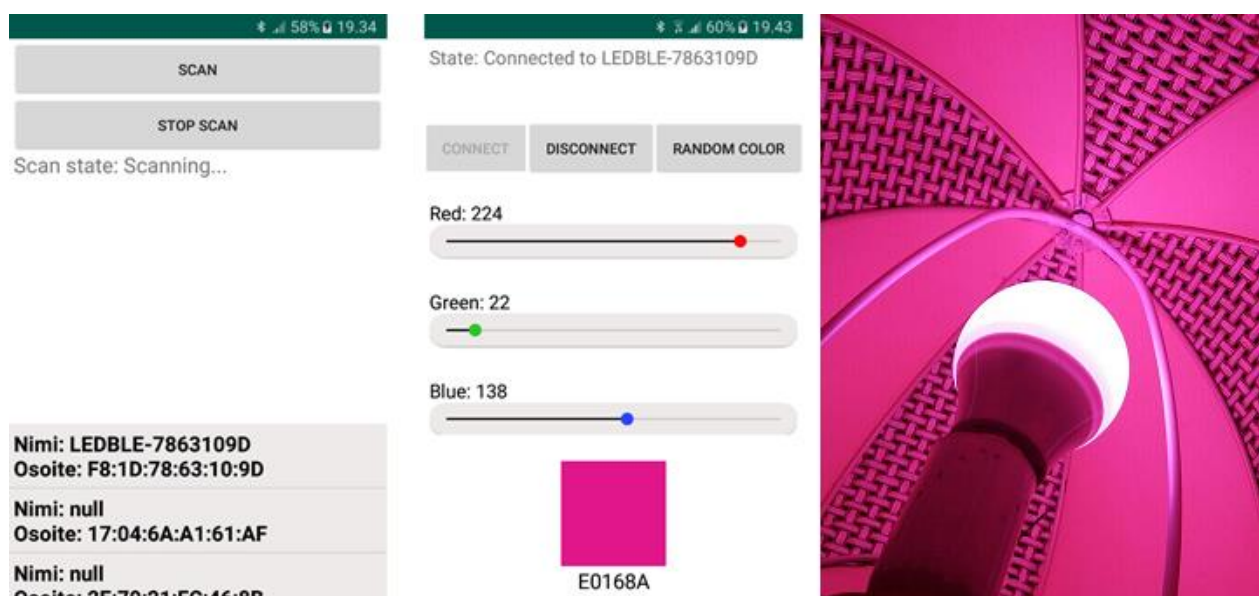
Lampulla on yksi characteristic jossa on notification ominaisuus mutta se ei lähettänyt mitään dataa jostain syystä. Pääsin kuitenkin testaamaan notificationin lukemista onnistuneesti toisella Bluetooth LE laitteella. Jatkokehitystä varten tämä on erittäin tärkeää saada toimimaan. Moni valmistaja kuitenkin asettaa monia characteristic arvoja, jotka he kuitenkin jättävät tyhjäksi eikä niillä tee mitään.

5.6 Lopullinen sovellus

IOT lamppusovellukseen ei ole tarpeellista hakea ja näyttää Bluetooth servicejä, characteristicjä tai mitään muuta, joka ei ole oleellista lampun värin säätämisen kannalta. Kaikki tekninen voidaan piilottaa tai poistaa ja varmistaa että sovellus toimii. Sovellus ei saa kaatua tai hidastua toimintakelvottomaksi missään vaiheessa.

Sovelluksen toiminnan päävaiheet ovat:

- Lähellä olevien Bluetooth LE laitteiden haku, tämän haun kesto voidaan ajastaa ja säädinkin sen 8 sekuntiin. Haku lopetetaan aina, jos sovellus sulkeutuu, haku lopetetaan manuaalisesti tai activity vaihtuu.
- Värin säätäminen Bluetooth LE lamppuun. Tämä tapahtuu nappia painamalla, joka säätää värin satunnaiseksi tai seekbar elementtejä muokkaamalla jolla voi säätää punaisen, vihreän ja sinisen arvon väliltä 0 ja 256 (Kuvio 18.)
- Yhteyden lamppuun voi avata ja sulkea nappien painalluksella ja sovellus ilmoittaa, jos yhteys katkeaa itsestään.



Kuvio 18. Sovelluksen pääikkunat sekä hehkulamppu.

6 LOPPUPÄÄTELMÄ PROJEKTISTA

Sovelluksen toteuttaminen onnistui hyvin muutamia vaikeuksia lukuun ottamatta. Valitettavaa oli huono dokumentaatio Bluetooth LE:n vähemmän käytetyistä ominaisuuksista, joista mainitsin kappaleessa 2.2. Kirjat tai internetfooromit ei antaneet käytännössä mitään tietoa joihinkin ominaisuuksiin Bluetooth LE:ssä joka ei ole niin yllättävää kun ottaa huomioon miten vähän ne on käytössä tällä hetkellä.

6.1 Android Studio kehitysympäristönä

Android Studio on mielestäni loistava ohjelma vaikkakin käynnistyessä hieman hidas. Lisäksi ohjelmien ajaminen virtuaaliympäristössä simuloituilla laitteilla vaatii virtualisointiominaisuuden, jota ei ole kaikissa tietokoneissa tai tietokoneen ytimissä. Ohjelmien ajaminen toimii ihan hyvin omilla fyysisillä Android laitteilla lukuun ottamatta hitautta ensimmäistä kertaa sovellusta ajettaessa tai pieniä ongelmia laitteen tunnistuksessa, kun se on kiinnitetty tietokoneeseen USB-johdolla.

Ohjelmassa on pieniä virheitä kuten logi joka ei tunnista joka kerta onko sovellus auki vai ei, se kuitenkin korjautuu helposti sovelluksen sulkemisella ja uudelleenavaamisella. Silloin tällöin Android Studio ilmoitti odottamattomasta ongelmasta ja kaatui. Se on kuitenkin odotettua painavalta ja monimutkaiselta ohjelmalta ja se tapahtui hyvin harvoin. Uudelleenkäynnistys korjasi senkin eikä tallentamaton työ ikinä tuhoutunut. Kaiken kaikkiaan Android Studio on hyvä ohjelma, josta eniten tykkäsin ennakoivasta syötöstä ja tiedostojen selkeästä valikkorakenteesta.

6.2 Sovelluksen toimivuus

Projektissa tehty sovellus on luotettava eikä se kaadu ja jatkokehitysmahdollisuudet ovat lähes rajattomat. Sovellus tarkkailee yhteyttä ja toimii välittömästi yhteyden tilan mukaan. Jos yhteys onnistuu tai katkeaa, sovellus ilmoittaa välittömästi käyttäjälle ja tekee tarvittavat toimet sovelluksen kaatumisen ehkäisemiseksi. Sovellus on mahdollisimman yksinkertainen käyttää, joka vähentää riskiä

sovelluksen kaatumiseen tai hidastumiseen eikä sen käyttö vaadi opettelua. Lisäksi sovelluksesta on helppo muokata eri kokoinen eri laitteille kuten pienille ja isoille puhelimille sekä tableteille Android Studion työkalujen avulla.

Bluetooth LE yhteyden kanssa työskentely vaatii paljon datan visualisointia sekä testausta koska on oltava varma mikä on yhteyden tila tai onko esimerkiksi notification ilmoitus tyhjä vai pelkästään luettu virheellisesti. Helppoa työssä oli saada välitöntä palautetta Bluetooth LE laitteen ominaisuuksien lukemisesta ja kirjoittamisesta, jos kirjoitus on tehty virheellisesti, silloin lamppu ei yksinkertaisesti vaihda väriä ja vika on helppo löytää.

Sovellusta hieman muokkaamalla on siitä mahdollista rakentaa minkä tahansa Bluetooth LE laitteen kontrolli koska periaate on sama kaikissa Bluetooth LE profiileissa. Kun toimiva yhteys Bluetooth LE laitteeseen on saatu valmiiksi ja perusominaisuudet kuten kirjoittaminen, lukeminen ja notification ilmoitukset toimivat, on ovet avoinna minkä tahansa Bluetooth LE laitteen kanssa työskentelyyn ja uusien sovellusten tekoon.

LÄHTEET

Android. 2019. Bluetooth low energy overview. Viitattu 12.12.2019. <https://developer.android.com/guide/topics/connectivity/bluetooth-le>.

Android. 2019. Request App Permissions. Viitattu 12.12.2019. <https://developer.android.com/training/permissions/requesting.html#kotlin>

Android. 2019. Kotlin overview. Viitattu 2.1.2020. <https://developer.android.com/kotlin/overview>

App Developer Magazine. 2018. Why Kotlin language use is skyrocketing. Viitattu 16.1.2020. <https://appdeveloperomagazine.com/why-kotlin-language-use-is-skyrocketing/>

Educba. 2018. Java vs Kotlin. Viitattu 12.1.2020. <https://www.educba.com/java-vs-kotlin/>

Guirdham, O. 2019. Bluetooth Will Help IoT Enabled Medical Devices Grow At 29.9% A Year. Viitattu 20.11.2019. <https://menafn.com/1098124753/Bluetooth-Will-Help-IoT-Enabled-Medical-Devices-Grow-At-299-A-Year>

Harris, R. 2018. Why Kotlin language use is skyrocketing. Viitattu 29.12.2019. <https://appdeveloperomagazine.com/why-kotlin-language-use-is-skyrocketing/>

Hoffman, C. 2018. Bluetooth 5.0: What's Different, and Why it Matters. Viitattu 20.12.2019. <https://www.howtogeek.com/343718/whats-different-in-bluetooth-5.0/>

Hollander, D. 2018. The Expansion of the Connected Device Market. Viitattu 18.11.2019. <https://www.bluetooth.com/blog/3-connected-device-segments/>

Industry ARC. 2015. Burgeoning Bluetooth Low Energy (BLE) Based Internet of Things (IoT) Movement to Drive the Market at a CAGR of 29% between 2014 and 2020. Viitattu 20.1.2020. <http://industryarcblog.com/burgeoning-bluetooth-low-energy-market/>

Jackowski, K. 2019. Kotlin vs. Java: Which One You Should Choose for Your Next Android App (UPDATED). Viitattu 12.1.2020. <https://www.netguru.com/blog/kotlin-java-which-one-you-should-choose-for-your-next-android-app>

Kacherovska, D. 2019. How Secure Is the BLE Communication Standard?. Viitattu 22.1.2020. <https://dzone.com/articles/how-secure-is-the-ble-communication-standard>

Kotlin. A modern programming language that makes developers happier. Open source forever. Viitattu 12.1.2020.

https://kotlinlang.org/?_ga=2.265660295.920902352.1579573230-751564175.1579573230

Madhur Bhargava. 2017. IoT Projects with Bluetooth Low Energy. Birmingham, Iso-Britannia. Packt Publishing Limited. Viitattu 2.1.2020

Mathieu, A. WHAT IS BLUETOOTH LOW ENERGY? Viitattu 2.11.2019. <https://elainnovation.com/what-is-ble.html>

Nielsen, J. 2017. Bluetooth low-energy technology for IoT applications. Viitattu 17.11.2019. <https://www.powerelectronicsnews.com/news/bluetooth-low-energy-technology-for-iot-applications>

Pinterest pinkart.in. Tanbaby 4.5W Magic Blue Bluetooth Smartphone Control Multicolor Dimmable Led Light Bulb Iphone Appl Viitattu 20.1.2020. <https://www.pinterest.com.mx/pin/597571444279227867/>

Rayome, A. 2018. Why Kotlin is exploding in popularity among young developers. Viitattu 20.12.2019. <https://www.techrepublic.com/article/why-kotlin-is-exploding-in-popularity-among-young-developers/>

Samsung. 2016. Level up your fitness with Galaxy Fit. Viitattu 28.11.2019. <https://www.samsung.com/uk/wearables/galaxy-fit-r370/SM-R370NZSABTU/>

Saurel, S. 2019. 10 Reasons Why Java Is Still Worth Learning In 2020. Viitattu 20.12.2019. <https://medium.com/javarevisited/10-reasons-why-java-is-still-worth-learning-in-2020-86a65282f735>

Shaked, U. 2016. Reverse Engineering a Bluetooth Lightbulb. Viitattu 20.1.2020. <https://medium.com/@urish/reverse-engineering-a-bluetooth-lightbulb-56580fcb7546>

Summitdata. 2013. BLE Overview. Viitattu 16.10.2019. <http://www.summitdata.com/blog/ble-overview/>

Tendyck, H. 2018. Uuden sukupolven majakkatekniikka. Viitattu 11.10.2019. <http://www.elektroniikkalehti.fi/index.php/13-news/7698-uuden-sukupolven-majakkatekniikka>

Thornsby, J. 2019. Kotlin vs Java for Android: key differences. Viitattu 18.1.2020. <https://www.androidauthority.com/kotlin-vs-java-783187/>

Townsend, K., Cufí, C. & Davidson, R., Akiba. 2014. Getting Started with Bluetooth Low Energy. Newton. Newton, Massachusetts, USA. O'Reilly Media, Inc. Viitattu 28.11.2019.